

CLAIMS

1. N+1 P-P (parallel program, hereinafter "P-P") module based on a single machine environment, which comprises:

N+1 P-P branch programs module ($N \geq 1$), which is run by operating the N+1 P-P branch programs which have an object code independent structures, in the way of time division (without the need of time division operation under the structure of parallel computers), for making a transmission and consistency of the P-P data under the supports of three classes of sequence-net instructions (or subroutines) for reading P-P data, writing P-P data, and making P-P data consistency in said P-P branch programs; and

a managing program module, for supporting the suspension status, ready status, and running status of the P-P branch programs in response to information from the P-P branch programs.

2. The N+1 P-P module of claim 1, further comprising a suspension-processing module, for processing the suspension status, ready status and running status of the current P-P branch programs.
3. The N+1 P-P module of claim 1, further comprising a P-P entering & exit management module, for initializing the P-P for each application and processing the exit of the P-P in response to information from a time-division- managing module.
4. The N+1 P-P module of claim 1, wherein the N+1th P-P branch program of P-P module executes a P-P data sequence, which is represented by a data consistency operation.
5. The N+1 P-P module of claim 4, wherein a token consistency operation of N+1th branch program corresponds to the P-P data, which is made consistency one by one directly by the control of the N+1th program, wherein if the P-P data has been written by other P-P branch program, then the consistency of the P-P data is executed, otherwise the N+1th branch program is in suspension status, after the execution of the P-P data's consistency, the P-P data read by other P-P branch program is valid.

6. The N+1 P-P module of claim 5, wherein, when P-Ps are embed and child P-Ps are called, the call-permission of the child P-P and the P-P data sequence are sequenced in the N+1th branch program.
7. The N+1 P-P module of claim 5, wherein, the N+1 programs include three kinds of P-P instructions (or subroutines) for reading the data, writing the data, and making the data in consistency respectively, and the P-P instructions (or subroutines) have the capability for setting, detecting, and processing the tokens, and also have the capability for supporting the transmissions of P-P data and the synchronization of P-P branch programs.
8. An operational method of N+1 P-P branch programs ($N \geq 1$) based on single machine environment, which comprises the following procedures:
 - initializing the N+1 P-P branch programs,
 - determining whether the P-P is terminated, and checking the ON/OFF switch flag of each P-P branch program, wherein:
 - if a switch flag of all P-P branch programs is OFF, which indicates the P-P is terminated, the record for showing the termination of the P-P and the manner for connecting with external programs are processed,
 - once the switch flag of one P-P branch program is ON, the suspension status for branch programs is checked, wherein,
 - if the check result is that all P-P branch programs are in suspension status, which indicates that though the execution of some P-P branch programs is not finished, the P-P branch programs can not be executed,
 - finding out the reasons that the P-P branch programs can not be executed,
 - processing the exiting of the P-P branch programs based on the reasons,
 - if the check result is that more than one P-P branch program is in ready status, enter the P-P branch programs queuing modules,
 - selecting a P-P branch program which is in ready status,
 - selecting a P-P branch program, loading its parameters, and
 - running the P-P branch program which is selected.
9. The N+1 P-P operation method of Claim 8, wherein, the initializing procedure comprises the loading of parameters of P-P branch programs and the resetting of the flag zone the P-P.

10. The N+1 P-P operation method of Claim 9, wherein, the loading of the parameters of the P-P branch program comprises:
- loading the flag ON for the N+1 branch programs of the P-Ps,
 - setting an entry address for each P-P branch programs,
 - setting a data initial values for respective registers, and
 - resetting a P-P flag zone and a data flag of the P-P branch program.
11. The N+1 P-P operation method of Claim 8, wherein the P-P branch program comprises a subroutine for writing data, which comprises the following procedures:
- performing the writing operation for a P-P data,
 - checking the flag "had been invalid for consistency" of the P-P data,
 - if the flag is valid (indicating that the P-P data had been made the data consistency, as the consistency flag is invalid, the consistency operation had not been operated actually to make the consistency of the P-P branch program entering into the suspension status),
 - the status bit of N+1th P-P branch program of the ready/suspension flag of P-P is changed from the suspension status to the ready status,
 - if the flag is invalid, no specific operation is executed,
 - establishing the flag "consistency valid" and allowing the consistency P-P branch program to perform the consistency operation to the data, and
 - exiting write data subroutine.
12. The P-P operation method of Claim 8, wherein the P-P branch program comprises a consistency data subroutine, which comprises the following procedures:
- checking the "consistency valid" flag of this data, wherein,
 - if the "consistency valid" flag of this data is invalid, the P-P branch program is suspended,
 - setting the "had been invalid in consistency" flag,
 - proceeding the suspension process of the current N+1th branch program, saving the current running situation of the N+1th P-P branch program to be used when returning the P-P branch program,

exiting the P-P branch program, and keeping the P-P branch program in suspension status and quitting from the returning port, in order to re-select a new P-P branch program,

if the "consistency valid" flag of this data is valid, the consistency operation between the data and the token is performed to:

check the N bit "had been invalid in consistency" flag, and

return to the subroutine of data consistency.

13. The P-P operation method of Claim 12, wherein in the step of checking the N bit "had been invalid in consistency" flag, if the "had been invalid in consistency" flag is valid, the corresponding P-P branch program is switched from the suspension status to the ready status based on the content in the "had been invalid in consistency" flag.
14. The P-P operation method of Claim 8, wherein the P-P branch program comprises a read data subroutine, which comprises the following procedures:
 - checking the "read valid" flag of the data,
 - if the "read valid" flag of the data is valid,
 - then reading the data, and
 - returning the read data subroutine.
15. The P-P operation method of Claim 14, wherein the step for checking the "read valid" flag of the data further comprises:
 - if the "read valid" flag is invalid, then setting N bit "had been invalid in consistency" flags which correspond to different P-P branch programs;
 - the current P-P branch program is subjected to suspension process, and the current running status of the P-P branch program is saved so as to continue to run by switching from the suspension point when the P-P branch program returns; and
 - exiting the P-P branch program, switching to the suspension entrance of the P-P branch program, re-selecting a new P-P branch program.